PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-259452
(43)Date of publication of application : 24.09.1999

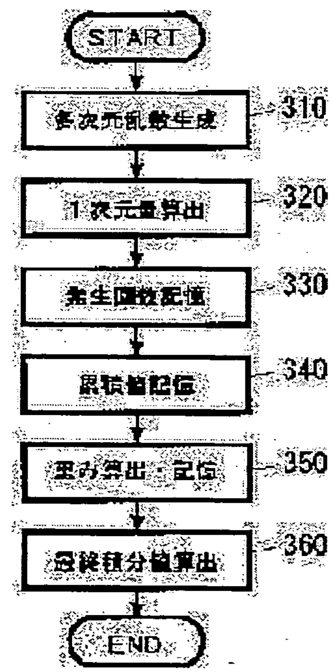
(51)Int.CI. G06F 17/10

(21)Application number : 10-034438
(22)Date of filing : 17.02.1998

(71)Applicant : INTERNATL BUSINESS MACH CORP <IBM>
(72)Inventor : MIZUTA HIDEYUKI

(54) FAST INTEGRATION METHOD AND SYSTEM

(57)Abstract:
PROBLEM TO BE SOLVED: To provide integration system/method effective even for highly dimensional Monte Carlo calculation, and high in convergence speed without impairing calculation time.
SOLUTION: One-dimensional quantity is calculated from multidimensional random numbers. It is desirable that one-dimensional quantity is adjusted to the feature of a function to be integrated. One-dimensional quantity is classified into several groups by a value and the number of generation times is recorded at every group. The accumulation value of the value of the function to be integrated, which corresponds to the multidimensional random number at that time, is recorded in accordance with previous grouping. The operation is repeated for N times and a ratio between the distribution of one-dimensional quantity, which is logically predicted for the respective groups, and the recorded number of generation times is recorded as weight for the group. The product of the accumulation value of the integrated function and weight is calculated at every group and the sum of all the groups is taken. The sum is divided by the sum of weight against all the groups. Then, an integration value by corrected Monte Carlo analysis is given.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12) 公開特許公報（Ａ）　(11)特許出願公開番号

特開平11－259452

(43)公開日　平成11年(1999) 9月24日

(51)Int.Cl.⁶　識別記号　ＦＩ
Ｇ０６Ｆ　17/10　　Ｇ０６Ｆ　15/31　Ｚ

審査請求　未請求　請求項の数10　ＯＬ　（全 10 頁）

(21)出願番号　特願平10－34438

(22)出願日　平成10年(1998) 2月17日

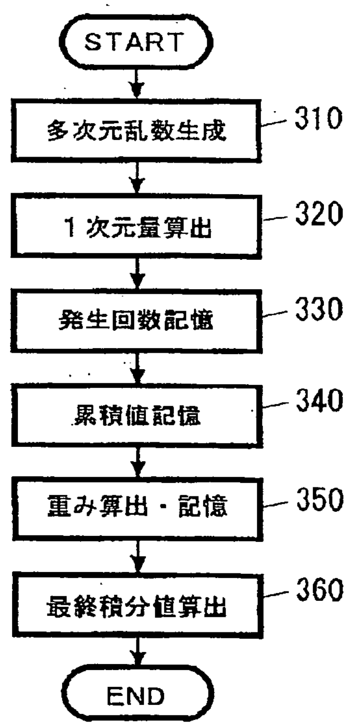
(71)出願人　390009531
インターナショナル・ビジネス・マシーンズ・コーポレイション
ＩＮＴＥＲＮＡＴＩＯＮＡＬ　ＢＵＳＩＮＥＳＳ　ＭＡＳＣＨＩＮＥＳ　ＣＯＲＰＯＲＡＴＩＯＮ
アメリカ合衆国10504、ニューヨーク州アーモンク　（番地なし）

(72)発明者　水田　秀行
神奈川県大和市下鶴間1623番地14　日本アイ・ビー・エム株式会社　東京基礎研究所内

(74)代理人　弁理士　坂口　博　（外１名）

(54)【発明の名称】　高速積分方法及びシステム

(57)【要約】
【課題】高次元のモンテカルロ計算にも有効な、計算時間を犠牲としない、収束速度の速い、積分システム及び方法を提供することである。
【解決手段】上記課題を解決するために、
・多次元数列を１次元量に変換する。
・モデルに応じて特徴を捉えて最適化する。
・乱数そのものを修正するのではなく重みづけ平均によって補正する。
・ヒストグラムの作成により乱数列の分布形状全体の補正を行う。
・ヒストグラム作成時のグループ毎に累積計算する。
これらの手段により、計算時間を犠牲とせずに収束の速度を増すことが可能となる。



【特許請求の範囲】
【請求項１】コンピュータにより多次元関数を高速に積分する積分システムであって、上記多次元関数を積分するための多次元の乱数の組みを複数生成する手段と、上記複数の乱数の組みのヒストグラム及び被積分関数の累積値を計算し、記憶装置に記憶する手段と、上記ヒストグラム及び上記累積値を用いて加重平均を計算する手段と、を具備することを特長とする、積分システム。
【請求項２】コンピュータにより多次元関数を高速に積分する積分システムであって、多次元の乱数の組みにより、多次元被積分関数の値を複数得る手段と、上記多次元乱数の組み毎に、１次元量を計算する手段と、上記１次元量を、値によって複数のグループに分類し、各グループ別に発生回数を記憶装置に記憶する手段と、上記多次元乱数に対応する被積分関数の値を、上記グループ別に累積値を記憶装置に記憶する手段と、上記発生回数の分布を利用して、上記発生回数と上記累積値の積の和から、加重平均を求める手段、を具備することを特長とする、積分システム。
【請求項３】コンピュータにより多次元関数を高速に積分する積分システムであって、多次元の乱数の組みにより、多次元被積分関数の値を複数得る手段と、上記多次元乱数の組み毎に、１次元量を計算する手段と、上記１次元量を、値によって複数のグループに分類し、各グループ別に発生回数を記憶装置に記憶する手段と、上記多次元乱数に対応する被積分関数の値を、上記グループ別に累積値を記憶装置に記憶する手段と、上記各グループに対して理論的に予想される１次元量の分布と記憶された上記発生回数の比を、グループに対する重みとして記憶装置に記憶する手段と、上記各グループに対して被積分関数の累積値と重みの積を計算する手段と、全グループについて上記積の和を計算する手段と、上記和を、全グループに対する重みの和によって除算する手段と、を具備することを特長とする、積分システム。
【請求項４】コンピュータにより多次元関数を高速に積分する積分方法であって、上記多次元関数を積分するための多次元の乱数の組みを複数生成する段階と、上記複数の乱数の組みのヒストグラム及び被積分関数の累積値を計算し、記憶装置に記憶する段階と、上記ヒストグラム及び前記累積値を用いて加重平均を計算する段階と、を有することを特長とする、積分方法。
【請求項５】コンピュータにより多次元関数を高速に積分するためのプログラムを含む媒体であって、該プログラムが、上記多次元関数を積分するための多次元の乱数の組みを複数生成する機能と、上記複数の乱数の組みのヒストグラム及び被積分関数の累積値を計算し、記憶装置に記憶する機能と、上記ヒストグラム及び前記累積値を用いて加重平均を計算する機能と、
を有することを特長とする、プログラムを含む媒体。
【請求項６】コンピュータにより、金融派生商品の価格計算を高速に行う、金融派生商品の価格計算システムであって、多次元関数を積分するための多次元の乱数の組みを複数生成する手段と、上記複数の乱数の組みのヒストグラム及び被積分関数の累積値を計算し、記憶装置に記憶する手段と、上記ヒストグラム及び上記累積値を用いて加重平均を計算する手段と、を具備することを特長とする、金融派生商品の価格計算システム。
【請求項７】コンピュータにより、金融派生商品の価格計算を高速に行う、金融派生商品の価格計算システムであって、多次元の乱数の組みにより、多次元被積分関数の値を複数得る手段と、上記多次元乱数の組み毎に、１次元量を計算する手段と、上記１次元量を、値によって複数のグループに分類し、各グループ別に発生回数を記憶装置に記憶する手段と、上記多次元乱数に対応する被積分関数の値を、上記グループ別に累積値を記憶装置に記憶する手段と、上記発生回数の分布を利用して、上記発生回数と上記累積値の積の和から、加重平均を求める手段、を具備することを特長とする、金融派生商品の価格計算システム。
【請求項８】コンピュータにより、金融派生商品の価格計算を高速に行う、金融派生商品の価格計算システムであって、多次元の乱数の組みにより、多次元被積分関数の値を複数得る手段と、上記多次元乱数の組み毎に、１次元量を計算する手段と、上記１次元量を、値によって複数のグループに分類し、各グループ別に発生回数を記憶装置に記憶する手段と、上記多次元乱数に対応する被積分関数の値を、上記グループ別に累積値を記憶装置に記憶する手段と、上記各グループに対して理論的に予想される１次元量の分布と記憶された上記発生回数の比を、グループに対する重みとして記憶装置に記憶する手段と、上記各グループに対して被積分関数の累積値と重みの積を計算する手段と、全グループについて上記積の和を計算する手段と、上記和を、全グループに対する重みの和によって除算する手段と、を具備することを特長とする、金融派生商品の価格計算システム。
【請求項９】コンピュータにより、金融派生商品の価格計算を高速に行う、金融派生商品の価格計算方法であって、多次元関数を積分するための多次元の乱数の組みを複数生成する段階と、上記複数の乱数の組みのヒストグラム及び被積分関数の累積値を計算し、記憶装置に記憶する段階と、上記ヒストグラム及び前記累積値を用いて加重平均を計算する段階と、を有することを特長とする、金融派生商品の価格計算方法。
【請求項１０】コンピュータにより金融派生商品の価格計算を高速に行うためのプログラムを含む媒体であって、該プログラムが、多次元関数を積分するための多次元の乱数の組みを複数生成する機能と、上記複数の乱数の組みのヒストグラム及び被積分関数の累積値を計算し、記憶装置に記憶する機能と、上記ヒストグラム及び前記累積値を用いて加重平均を計算する機能と、

を有することを特長とする、プログラムを含む媒体。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】本願は、高速に積分を行うシステム及び方法に関し、特に多次元数列を１次元量に変換して、重みづけ平均によって補正することを特長とする、高速積分システム及び方法に関する。

【０００２】

【従来の技術】従来数値積分を行う方法には、擬似乱数によるモンテカルロ法の分散減少法として代表的なものに以下にあげるようなものが知られている。

対称変数法

乱数列に対して、絶対値は等しく反対の符号を有する数列を付け加えることによって、乱数の分布を対称なものとし、分散を減少させようとするもの。

モーメント照合法

発生させた乱数列より、平均および分散を求め、それが本来あるべき値となるように乱数の値自体を修正するもの。それぞれ、発生させた乱数列自体に手を加えるため、自己相関の問題がある。さらにモーメント照合法はそれまでに発生させた全乱数を記憶しておく必要があり、高次元では非実用的である。また、ＬＤＳ（low-discrepancy sequence）のように偏りのないサンプル点を与えるものに対して、点の位置自体を変更するこれらの方法は有効では無い。

【０００３】

【発明が解決しようとする課題】従って、本発明が解決しようとする課題は、高次元のモンテカルロ計算にも有効な積分システム及び方法を提供することである。また別の課題は、積分モデルに応じて特徴を捉えて最適化することのできる、積分システム及び方法を提供することである。また別の課題は、ＬＤＳ（low-discrepancy sequence）に対して効果的な分散減少法としての、積分システム及び方法を提供することである。また別の課題は、乱数列の分布形状全体の補正を行うことができる、積分システム及び方法を提供することである。また別の課題は、積分計算に際し、多量のメモリを必要としない、積分システム及び方法を提供することである。

【０００４】

【課題を解決するための手段】上記課題を解決するために、

・多次元数列を１次元量に変換する。

・モデルに応じて特徴を捉えて最適化する。

・乱数そのものを修正するのではなく重みづけ平均によって累積計算を補正する。

・ヒストグラムの作成により乱数列の分布形状全体の補正を行う。

・ヒストグラム作成時のグループ毎に累積計算する。

これらの手段により、計算時間を犠牲とせずに収束の速度を増すことが可能となる。

【０００５】

【発明の実施の形態】多次元の乱数より、多次元被積分関数の値が得られるものとする。ここで１つの値を得るのに要する乱数の組みを１シナリオと呼ぶ。この操作を、Ｎ回繰り返し、得られたＮ個の値の算術平均を求めることにより、関数の積分を行うのが本来のモンテカルロ法である。ここで各操作において、多次元乱数より１次元量を計算する。この１次元量は、被積分関数の特徴にあったものであるほうが好ましい。この１次元量を、値によっていくつかのグループに分類し、各グループ毎に発生回数を記録する。また、このときの多次元乱数に対応する被積分関数の値も、先のグループ分けにしたがって累積値を記録する。この操作をＮ回繰り返したのち、各グループに対して理論的に予想される１次元量の分布と記録された発生回数の比を、そのグループに対する重みとして記録する。たとえば、多次元乱数として正規分布に従うものをとり、１次元量を各成分の和とすると、これは再び正規分布に従うため、その理論分布は容易に計算する事ができる。最後に、各グループに対して被積分関数の累積値と重みの積を計算し、すべてのグループについての和をとる。この和を、すべてのグループに対する重みの和によって除することにより、修正されたモンテカルロ法による積分値が与えられる。

【０００６】

【実施例】以下、図面を参照して本発明の実施例を説明する。図１には、本発明において使用される積分システムのハードウェア構成の一実施例を示す概観図が示されている。システム１００は、中央処理装置（ＣＰＵ）１とメモリ４とを含んでいる。ＣＰＵ１とメモリ４は、バス２を介して、補助記憶装置としてのハードディスク装置１３（またはMO、ＣＤ－ＲＯＭ２３、ＤＶＤ等の記憶媒体駆動装置）とＩＤＥコントローラ２５を介して接続してある。同様にＣＰＵ１とメモリ４は、バス２を介して、補助記憶装置としてのハードディスク装置３０（またはMO２８、ＣＤ－ＲＯＭ２３、ＤＶＤ等の記憶媒体駆動装置）とＳＣＳＩコントローラ２７を介して接続してある。フロッピーディスク装置２０はフロッピーディスクコントローラ１９を介してバス２へ接続されている。好ましくはハードディスク装置１３若しくはメモリ４内に、各グループ別に発生回数、多次元乱数に対応する被積分関数の値、グループ別の累積値などが記憶される。

【０００７】フロッピーディスク装置２０には、フロッピーディスクが挿入され、このフロッピーディスク等やハードディスク装置１３（またはMO、ＣＤ－ＲＯＭ、ＤＶＤ等の記憶媒体）、ＲＯＭ１４には、オペレーティングシステムと協働してＣＰＵ等に命令を与え、本発明を実施するためのコンピュータ・プログラムのコード若しくはデータを記録することができ、メモリ４にロードされることによって実行される。このコンピュータ・プ

ログラムのコードは圧縮し、または、複数に分割して、複数の媒体に記録することもできる。

【０００８】システム１００は更に、ユーザ・インターフェース・ハードウェアを備え、入力をするためのポインティング・デバイス（マウス、ジョイスティック等）７またはキーボード６や、視覚データをユーザに提示するためのディスプレイ１２を有することができる。また、パラレルポート１６を介してプリンタを接続することや、シリアルポート１５を介してモデムを接続することが可能である。このシステム１００は、シリアルポート１５およびモデムまたは通信アダプタ１８(イーサネットやトークンリング・カード)等を介してネットワークに接続し、他のコンピュータ等と通信を行うことが可能である。またシリアルポート１５若しくはパラレルポート１６に、遠隔送受信機器を接続して、赤外線若しくは電波によりデータの送受信を行うことも可能である。

【０００９】スピーカ２３は、オーディオ・コントローラ２１によってＤ／Ａ（デジタル／アナログ変換）変換された音声信号を、アンプ２２を介して受領し、音声として出力する。また、オーディオ・コントローラ２１は、マイクロフォン２４から受領した音声情報をＡ／Ｄ（アナログ／デジタル）変換し、システム外部の音声情報をシステムにとり込むことを可能にしている。

【００１０】このように、本発明の積分システムは、通常のパーソナルコンピュータ（ＰＣ）やワークステーション、ノートブックＰＣ、パームトップＰＣ、ネットワークコンピュータ、コンピュータを内蔵したテレビ等の各種家電製品、通信機能を有するゲーム機、電話、ＦＡＸ、携帯電話、ＰＨＳ、電子手帳、等を含む通信機能有する通信端末、または、これらの組合せによって実施可能であることを容易に理解できるであろう。ただし、これらの構成要素は例示であり、その全ての構成要素が本発明の必須の構成要素となるわけではない。

【００１１】図２に、本発明のブロック構成図を示す。まずブロック２１０は、充分な数のシナリオについてヒストグラムと被積分関数の累積値を得る、ヒストグラム作成及び累積計算ブロックである。このヒストグラムの作成により乱数列の分布形状全体の補正が可能になる。次にブロック２２０は、ヒストグラムを元に累積計算に補正を行い平均を求める、加重平均ブロックである。すなわち乱数そのものを修正するのではなく重みづけ平均によって累積計算を補正する。

【００１２】図３に、本発明のより詳細なブロック構成図を示す。図２のブロック２１０は図３のブロック３１０～３４０に相当する。また図２のブロック２２０は図３のブロック３５０～３６０に相当する。まずブロック３１０は多次元の乱数を生成する多次元乱数生成ブロックである。またブロック３２０は、この多次元乱数より１次元量を計算する、１次元量算出ブロックである。この１次元量は、被積分関数の特徴にあったものであるほうが好ましい。そしてブロック３３０はこの１次元量を、値によっていくつかのグループに分類した後、各グループ毎に発生回数を記憶する、発生回数記憶ブロックである。次にブロック３４０は、このときの多次元乱数に対応する被積分関数の値を、先のグループ分けに従い累積値を記憶する、累積値記憶ブロックである。次にブロック３５０は、各グループに対して理論的に予想される１次元量の分布と記録された発生回数の比を、そのグループに対する重みとして記憶する、重み算出記憶ブロックである。これにより、多次元乱数として正規分布に従うものをとり、１次元量を各成分の和とすれば、これは再び正規分布に従うので、その理論分布は容易に計算する事が可能になる。そしてブロック３６０は、最終積分値算出ブロックである。具体的には、まず各グループに対して被積分関数の累積値と重みの積を計算し、すべてのグループについての和をとる。この和を、すべてのグループに対する重みの和によって除することにより、求める積分値が得られる。

【００１３】図４及び図５に本発明のフローチャートを示す。より分かりやすく説明するために、金融派生商品の時価計算を例にとり説明する。まずステップ４１０で、多次元の乱数列を生成する。次にステップ４２０で多次元乱数より１次元量を計算するにあたり、各乱数の総和を求める。この１次元量は、被積分関数の特徴にあったものであれば他の実施例でもよい。ステップ４３０で、この１次元量を、値によっていくつかのグループに分類するための配列のインデックスとなる自然数に直す。そしてブロック４４０で乱数列のシナリオから価格を１つ決める。次にステップ４５０で、各グループ毎に発生回数及び、このときの多次元乱数に対応する被積分関数の値を、先のグループ分けにしたがって累積値を記憶する。そしてステップ４６０で充分な数のシナリオについてヒストグラムと価格の和を求めたかどうかが判断される。もしその結果がＹｅｓであれば、ステップ４７０でインデックスの最大値をｍａｘ ｉｎｄｅｘとし、処理は図５の加重平均を行うステップへと移る。もしその結果がＮｏであれば処理は再びステップ４１０へ戻る。なおｍａｘ ｉｎｄｅｘは予め定数として定義しておいてもよいし、処理中に動的に変更してもよい。

【００１４】ステップ５１０は、一次元量のインデックス(ノルムインデックス)をｉ、重み付け価格をｗｐｒｉｃｅ、全重みをｗｓｕｍとして初期化を行う。次にステップ５２０で、各グループに対して理論的に予想される１次元量の分布値を求める。そしてステップ５３０で、各グループに対して理論的に予想される１次元量の分布と記録された発生回数の比を、そのグループに対する重みとして記憶する。そしてステップ５４０で、重み付け価格ｗｐｒｉｃｅ、全重みｗｓｕｍを前記重みを掛けることにより算出する。そしてステップ５５０でインデックスｉを１だけ増加させる。ステップ５６０でインデッ

クスｉがｍａｘｉｎｄｅｘより大きいかを判断し、もしその結果がＮｏであれば処理は再びステップ５２０へ戻る。もしその結果がＹｅｓであれば、ステップ５７０で、被積分関数の累積値と重みの積のすべてのグループについての和を、すべてのグループに対する重みの和によって除する。これにより、修正されたモンテカルロ法による積分値が与えられる。

【００１５】本発明を用いた複雑な金融派生商品の価格計算の例を示す。従来は、金利変動を考慮した価格プロセスの期待値を求める際、擬似乱数によるモンテカルロ法がしばしば用いられた。しかし、巨大な資産のリスク管理や、顧客に対するリアルタイムデモンストレーションにおいては、まだまだ高速化が求められる。金利変動 dr を、時間経過 dt とウィーナー過程 dz によって与える。Mean reversiona, b と drift μ および volatility σ は定数である。Vasicek Model を用いると

【数１】

$$dr = a(b-r)dt + \sigma dz$$

次に Euler 近似によって離散化を行う。時間間隔を Δt、N(0,1) の正規乱数をu_i $(i=0,\ldots,i_{max})$ とすると、漸化式

【数２】

$$r_{i+1} - r_i = a(b - r_i)\Delta t + \sigma u_i\sqrt{\Delta t}$$

によって、時刻 $t=i\Delta t$ の金利 r_i が与えられる。ここで金利変動により価格が決定される商品の例として、５年割引債を考える。5年後に1単位支払われる債券の現在価値を求める。現在価値は Vasichek Model によって与えられる5年間の金利変動に従って、割り引いて考えなくてはならない。これは、その現在価値分の変動金利のもとでの預金が、５年後に支払われる債券の額面金額に丁度等しくなるとも考えられる。離散化単位 Δt を 1/288 年とすると、i_{max} = 1439 となる。このとき、各時点の金利 r_i を用いて、価格 P は

【数３】

$$P = \exp\left(-\Delta t \sum_{i=0}^{i=i_{max}} r_i\right)$$

で与えられる。よって、例えば、

r_0 = 0.021673, a = 0.1817303, b = 0.0825398957, σ = 0.0125901

として先の漸化式をといて、r_1 から r_{1439} を与えれば、５年割引債の現在価値が計算できる。モンテカルロ法では、金利を求める漸化式において、擬似乱数によりu_1からu_{1439}を与え、それらによる価格 P を求める。この計算は、次々と乱数を発生させる毎に異なった P を与えるため、それぞれf_i として、N 個の価格を計算し、その平均値を結果とする。

【数４】

$$\hat{f} = \frac{1}{N}\sum_{i=1}^{N} f_i$$

N を増やしていくと、評価誤差は $1/\sqrt{N}$ の比率で減少していく。このため 10倍の精度を得るためには、100倍の計算が必要となる。f_i を求めるために用いるサンプルの抽出に、擬似乱数ではなくLow-discrepancy sequence を用いることで、評価誤差を 1/N とできる。これによって、必要な精度を得るのに要する計算量を大幅に減らすことができる。ここで、さらに本発明のヒストグラム法を用いれば、高速に収束が行われ、さらに精度が良くなる。例えばモンテカルロ法で１日を要する計算を、LDSでは１０分、本発明を併用したLDSでは１分で行えるようになる。

【００１６】図６、図７に U を1000次元の正規乱数とし、被積分関数に債券の簡単なモデルであるexp(0.01ΣUi)を用いた場合の従来の手法及び本発明の方法による場合の収束過程を図示する。まず図６は本発明の方法を用いない場合の収束過程である。ＬＤＳを用いた場合は擬似乱数より収束が早いことがわかる。しかしながら、サンプル数１００００回でも充分収束しているとは言えない。そこで図７に本発明の方法を用いた場合（ヒストグラム幅２及び３）の収束過程を示す。これからサンプル数１００００回の場合、本発明を使用しない場合と比較して充分高速に理論積分値に収束することがわかる。なお理論値からの相対誤差を図８の表に示す。

【００１７】参考として、図４及び図５のフローチャートに従ったプログラム例を以下に記載する。(なお STATNUM, STATLEN, TNUM, Dは予め 2000、2.0、100、1000等と定義しておいてもよいし、動的に変更しても構わない)

```
void gettheory(double theory[STATNUM]);
double getaverage(double theory[STATNUM],
int ustat[STATNUM], double psum[STATNUM], long cnt);
int
main()
{
        long            jj, maxcount = 10000;
        double          u[D];
        double          norm, price, priceave;
        int             i, nindex, ustat[STATNUM];
```

```
        double          psum[STATNUM], theory[STATNUM];
        for (i = 0; i < STATNUM; i++)
        {
                ustat[i] = 0;
                psum[i] = 0;
        }
        gettheory(theory);
        for (jj = 1; jj <= maxcount; jj++)
        {
                getrandom(u);       /* get D-dim random sequences */
                norm = calcnorm(u); /* calc norm of random sequence */
                nindex = (int)((norm+STATNUM*STATLEN / 2) / STATLEN);
                price = value(u);   /* get a price from a senario */
                ustat[nindex]++;
                psum[nindex] += price;
        }
        priceave = getaverage(theory, ustat, psum, maxcount);
        printf("result : %.16f¥n", priceave);
}
void gettheory(double theory[STATNUM])
{
        int mid = STATNUM / 2;
        int i, j;
        double dx = STATLEN / TNUM;
        double c, f, sum, x, total = 0;

        c = dx / sqrt(2 * D * 3.1415926536);
        for (i = 0; i < mid - 1; i++)
        {
                sum = 0;
                for (j = 0; j < TNUM; j++)
                {
                        x = i * STATLEN + (j + 0.5) * dx;
                        f = exp(- x * x / (2 # D)) * c;
                        sum += f;
                }
                theory[mid + i] = theory[mid - i -1] = sum;
                total += sum;
        }
        theory[0] = theory[STATNUM -1] = 0.5 - total;
}
double getaverage(double theory[STATNUM],
int ustat[STATNUM], double psum[STATNUM], long cnt)
{
        double pricesum = 0.0, wsum = 0.0, weight;
        int i;
        for (i = 0; i < STATNUM; i++)
        {
                weight = theory[i] / ustat[i];
                pricesum += psum[i] * weight;
```

```
                wsum += weight * ustat[i];
            }
            return pricesum / wsum;
        }
```

【００１８】

【発明の効果】オプションやスワップ等の複雑な金融派生商品の価格計算は、従来モンテカルロ法によって求められてきた。これは高性能な計算機を用いても非常に時間を要する。このような状況は乱数のかわりにLow-discrepancy sequance(LDS)を用いる事によって大幅に改善される。本発明はこれをさらに補正を加えて、必要な精度の価格を得るのに要する計算量を減らすことができる。たとえば、モンテカルロ法で１日を要する計算を、LDSでは１０分、本発明を併用したLDSでは１分で行えるようになる。多数の資産を持つ銀行でのリスク管理や客先での価格説明にはこのような高速な計算技術が不可欠である。本発明の方法を用いることにより、計算時間を犠牲とせずに収束の速度を増すことが可能となる。

【００１９】

【図面の簡単な説明】

【図１】本発明において使用される積分システムのハードウェア構成の概観図である。

【図２】本発明の積分システムのブロック構成図である。

【図３】本発明の積分システムのより詳細なブロック構成図である。

【図４】本発明のフローチャートを示す。

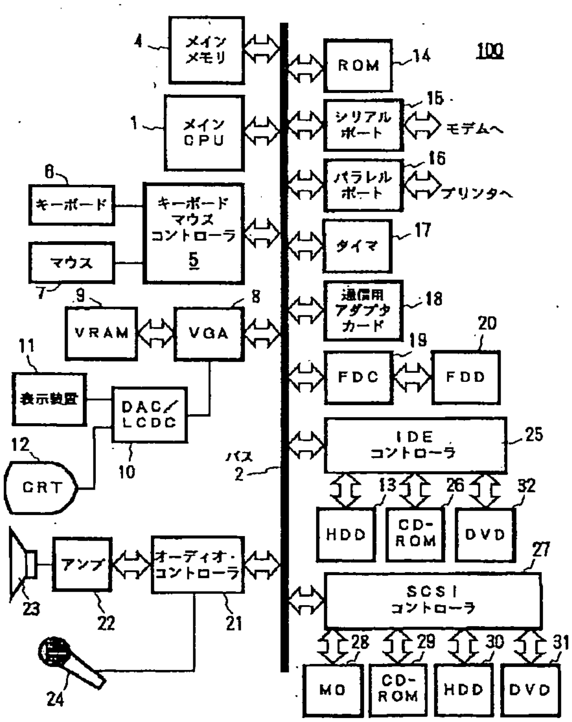
【図５】本発明のフローチャートを示す。

【図６】被積分関数に債券モデルを想定した場合の従来技術による収束過程を示す。

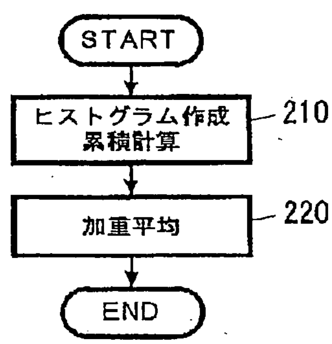
【図７】被積分関数に債券モデルを想定した場合の本発明による収束過程を示す。

【図８】本発明を使用する場合としない場合の理論値からの相対誤差を示す表である。

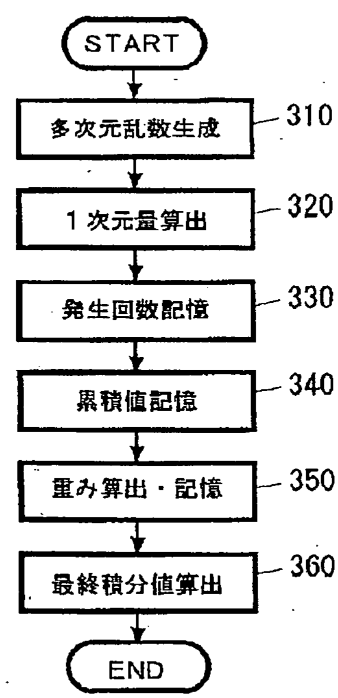
【図１】



【図２】



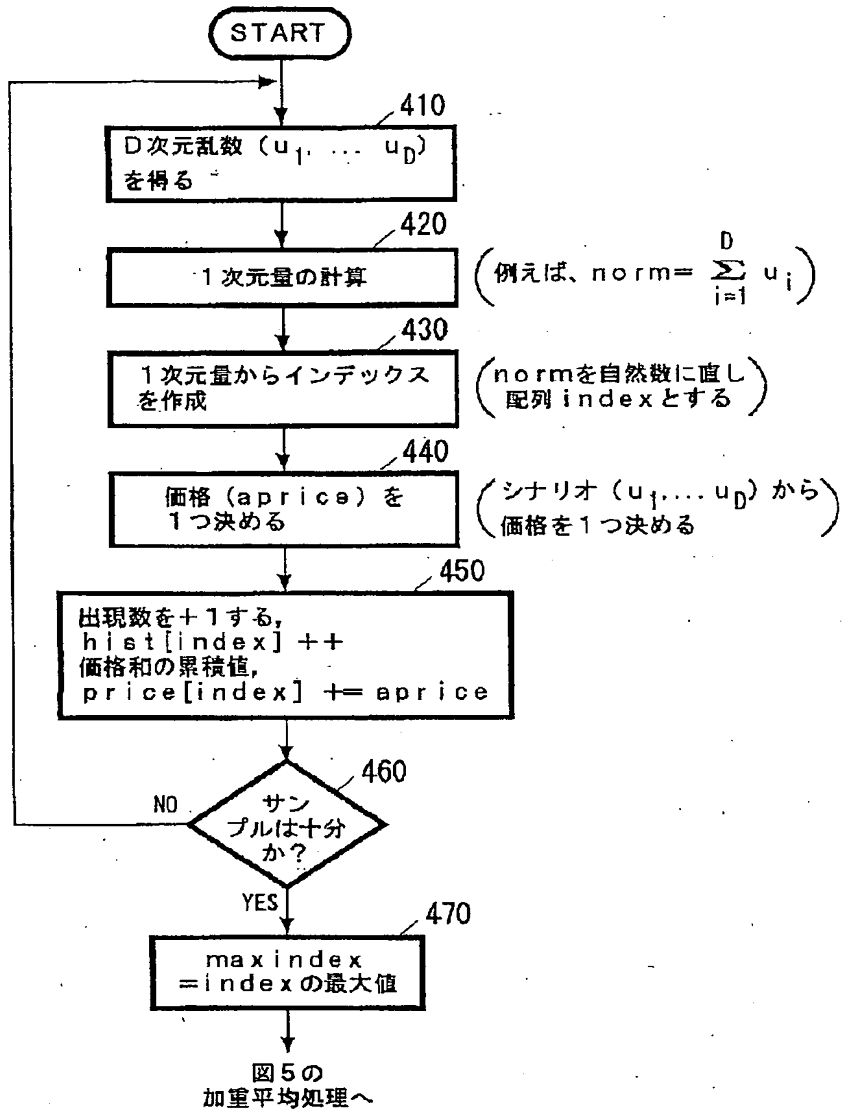
【図３】



【図８】

サンプル数 N	重みづけなし	重みづけあり
1,000	0.0007482	0.00005741
100,000	0.0000244	0.000008087

【図４】


START
410
D次元乱数（u1, ... uD）を得る
420
１次元量の計算
（例えば、norm＝Σ i=1 D ui）
430
１次元量からインデックスを作成
（normを自然数に直し配列indexとする）
440
価格（aprice）を１つ決める
（シナリオ（u1,... uD）から価格を１つ決める）
450
出現数を＋１する，
hist[index] ++
価格和の累積値，
price[index] += aprice
460
サンプルは十分か？
NO
YES
470
maxindex
＝indexの最大値
図５の
加重平均処理へ

【図5】

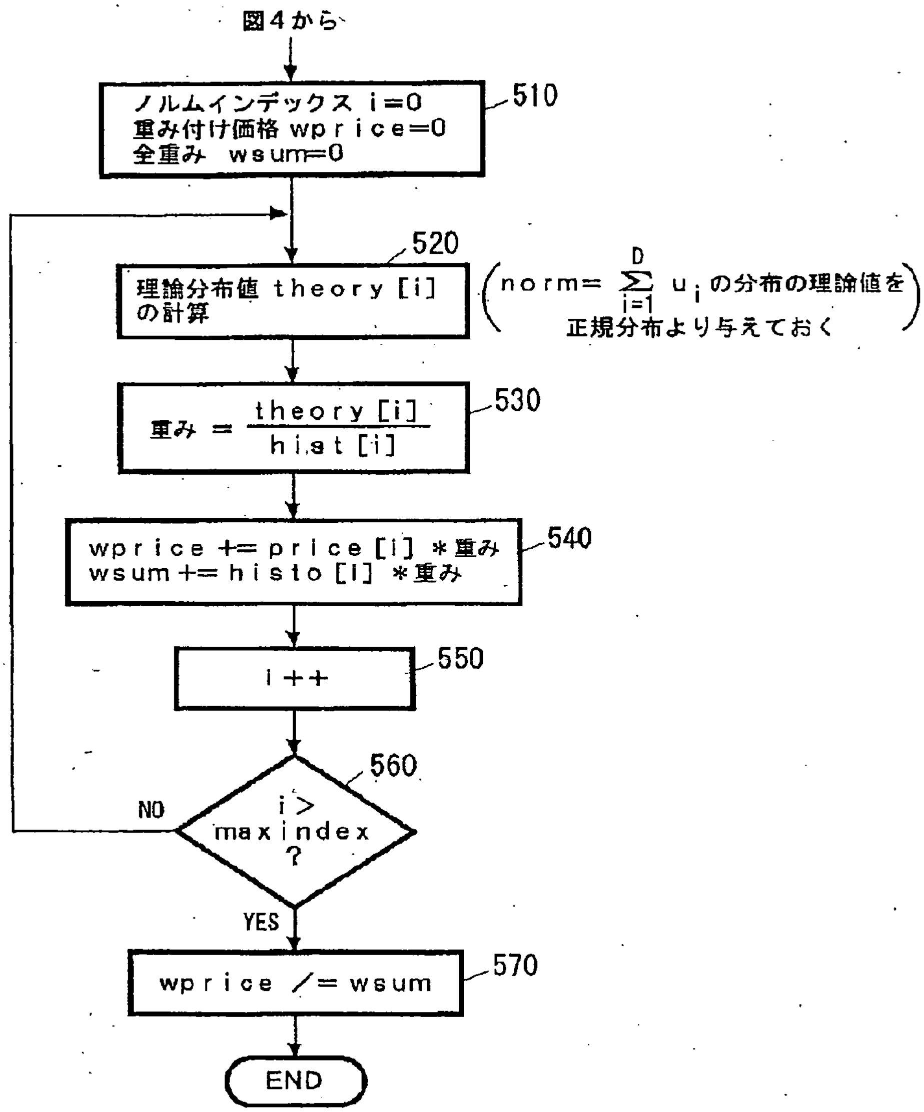
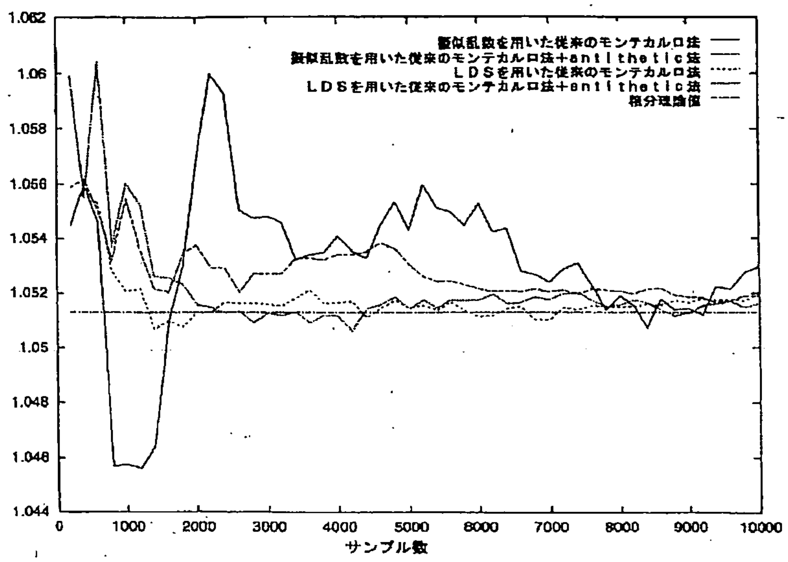
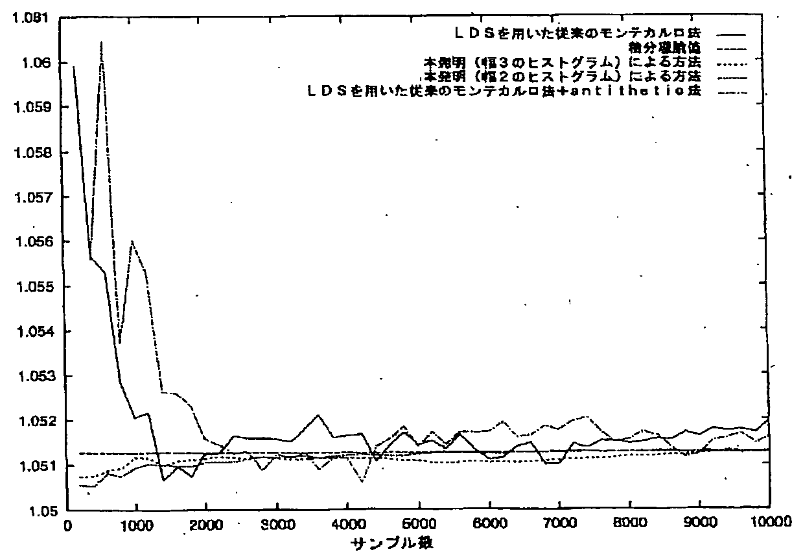


図4から
ノルムインデックス i=0
重み付け価格 wprice=0
全重み wsum=0
510
理論分布値 theory [i] の計算
520
(norm= Σ_{i=1}^{D} u_i の分布の理論値を正規分布より与えておく)
重み = theory [i] / hist [i]
530
wprice += price [i] *重み
wsum += histo [i] *重み
540
i++
550
i > maxindex ?
560
NO
YES
wprice /= wsum
570
END

【図6】


1.062
1.06
1.058
1.056
1.054
1.052
1.05
1.048
1.046
1.044
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
サンプル数
擬似乱数を用いた従来のモンテカルロ法
擬似乱数を用いた従来のモンテカルロ法+antithetic法
LDSを用いた従来のモンテカルロ法
LDSを用いた従来のモンテカルロ法+antithetic法
積分理論値

【図7】


1.061
1.06
1.059
1.058
1.057
1.056
1.055
1.054
1.053
1.052
1.051
1.05
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
サンプル数
LDSを用いた従来のモンテカルロ法
積分理論値
本発明（幅3のヒストグラム）による方法
本発明（幅2のヒストグラム）による方法
LDSを用いた従来のモンテカルロ法+antithetic法